PROJECT DESCRIPTIONS — MATHCAMP 2016

Contents

Assaf's Projects

Board Game Design/Analysis. (Assaf)

Description: I'd like to think about and design a board game or a card game that has interesting math, but can still be played by a non-mathematician.

I consider Set a good example, since it deals with finding lines in a vector space over a finite field, but can still be played by everyone.

In this project, we will play many board games, and discuss the ideas behind them. We will think about what happens when we change parameters, probabilities, or mechanics, and use this information to design and playtest our own board game.

Structure: Group activity with what-if scenarios as problem sets.

Expected Input: 3-4hr/week, lots of time in week 5 for playtesting.

Expected Output: A board game.

Difficulty:

Prerequisites: None.

Covering Spaces. (Assaf)

Description: Guided reading of the first chapter of Hatcher's Algebraic Topology. You will read about covering spaces, and prove that every subgroup of a free group is free.

Structure: Guided Reading

Expected Input: About 50 pages of math reading

Expected Output: Knowledge!

Difficulty:

Prerequisites: Topology, group theory.

Differential Geometry. (Assaf)

Description: A guided reading of DoCarmo's Riemannian Geometry. The book is about manifolds; I hope to get to the part where he talks about geodesics.

Structure: Guided reading

Expected Input: 70 pages of reading

Expected Output: Knowledge!

Difficulty:

Prerequisites: Linear algebra, group theory, topology, calculus.

Chris's Projects

Π_1 **of Campus.** (Chris)

Description: Want to study algebraic topology? And apply it—to the dorms?? In this project I will explain to you the definition of the fundamental group of a topological space, an invariant that is easy to explain but hard to calculate. Roughly speaking, it is the group of rubber closed rubber bands running in a space, where to rubber bands are considered the same if you can transform them into each other without leaving the space. Then it is up to you to think of ways of calculating the fundamental group of various spaces, so that you finally can determine the fundamental group of the space we live in—the Woodmann/Foss dorm area.

Structure: Independent study, you can check in with me if you're on the right track

Expected Input: less than 10 hours over the course of camp

Expected Output: a group, possible a huge one

Difficulty:

Prerequisites: Some notion of continuity, know what a group is

Don's Projects

Analytics of Football Drafts. (Don)

Description: In most professional sports, teams pick new players via a draft. They also sometimes trade those draft picks for other draft picks. However, nobody really knows what a fair trade looks like. In football, there's a chart based on past trades, but it's widely acknowledged that it values top picks too highly.

In this project, we'll generate a bunch of fake drafts, where the players conform to a reasonable valuation model, and where teams estimations of those values are off by a random amount. From this setup, we'll try to answer the question of what makes a trade fair.

Structure: You'll be creating the model mostly from scratch, and that will be a decent amount of work. Once we have the output, we'll meet a couple times a week to analyze it.

Expected Input: Some coding, some statistical analysis, and maybe organizing some real-time competitions.

Expected Output: Ideally, a paper describing the results that could be published/presented at a conference. At least, a cool model to present.

Difficulty: 🌶🌶

Prerequisites: Programming experience (such as Sam's Statistical Modeling class)

Open Problems in Knowledge Puzzles. (Don)

Description: In the Sum and Product Puzzles class, we met several open problems. Some of them, such as how many stable solutions there are to the classic Impossible Problem, are very hard to approach. Others, like how many puzzles involving only basic "I know" and "I don't know" statements have nontrivial solutions, are ripe for at least partial progress - and maybe some of them, we'll be able to show are equivalent to more famous open problems, like Goldbach's conjecture.

Structure: I have a lot of questions we could pursue using various means - some, through applications of number theory, some through programming, some just by trying a lot of ideas. I'll give you these questions, and you'll try to answer them.

Expected Input: Meeting a couple times a week to talk about your progress, and working several hours between each meeting.

Expected Output: A paper or poster summarizing the progress you make.

Difficulty: 🌶🌶

Prerequisites: Sum and Product Puzzles

Finite Topological Spaces. (Don + Yuval)

Description: Most topological spaces that mathematicians study are infinite, even uncountable. This might lead you to think that only infinite spaces are interesting. In fact, this could not be farther from the truth! In this project, well learn as much as we can about finite topological spaces. Well start by looking at examples of spaces with very small numbers of points to try and see what finite spaces can look like. Then well use these examples to figure out how to characterize all finite spaces (and continuous maps between them) with axioms that are much simpler than the general axioms for topological spaces. If you get far enough with this, we might even be able to learn a little about how finite spaces relate to algebraic topology.

Structure: You'll be doing most of the work; we'll guide you in productive directions and suggest ideas and definitions to think about. This project probably works best with two campers bouncing ideas off each other, although one camper can certainly make significant progress.

Expected Input: You should be ready to put in some amount of work every day; finite topological spaces are weird, and you'll be building your intuition for them from scratch.

Expected Output: As good an understanding of the behavior of finite spaces as we can get, and a poster about this if you want to make one.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Point-set topology

David's Projects

p-adics in Sage. (David)

Description: Sage is an open-source math software project (`www.sagemath.org`) aiming to be a free alternative to Mathematica, Maple, Matlab and Magma. It includes an implementation of the p-adic numbers, with various features missing. The project is to add some of these features.

Structure: I'll meet with you to select projects and to help answer questions about p-adics in Sage. You'll work in groups on new features.

Expected Input: We'll meet this week during TAU for 2-3 hours total. After that, I think you'll need at least 4 or 5 hours to implement something and get it through the review process; with more time you'll be able to do cooler things.

Expected Output: Working code, submitted to the Sage project.

Difficulty:

Prerequisites: Took Clifton's class or know about p-adics, know how to program in Python and (ideally) C.

Gloria's Projects

Mathematical Crochet and Knitting. (Gloria + Assaf + Jane)

Description: One of the best ways to visualize surfaces in three dimensions is to hold them in your hands and play with them. In this project, we'll make our own hyperbolic planes, Möbius strips, Klein bottles, Seifert surfaces, Lorenz manifolds and more, all out of yarn or felt or fabric! No previous crocheting, knitting, or sewing experience is necessary.

Structure: We'll have introductory "how to" sessions; after that, you can work on projects pretty much wherever and whenever you feel like it! Extra instruction will be needed for Kat Bordhi's Möbius cast on, and for grafting. Technical help will be available during TAU.

Expected Input: Totally up to you. Expect a few hours to learn the basics if you've never done any of these things before; after that, it's up to you.

Expected Output: Mathematical surfaces that you can hold and play with!

Difficulty:

Prerequisites: None.

Jackie's Projects

Hanabi AI. (Jackie)

Description: Can we teach a computer program to play the game Hanabi? There are many possible approaches. You can design your AI to play with itself (allowing you to create interesting and computationally intensive new strategies) or with real people (creating challenges of interpreting humans). You could even have the AI figure out what strategies and conventions are optimal!

Structure: You can work alone or in a group. We'll have an initial meeting to discuss potential approaches. After that, it'll be up to you how much you want to work on it, but you can always come talk with me for thoughts on strategy or debugging help.

Expected Input: Put in however much effort you want. Simulating basic Hanabi strategy will probably take more than 10 hours, and you could easily spend more time to design something more complex, or spend less time to make partial progress.

Expected Output: A Hanabi AI program! And lots of fun playing games of Hanabi to practice.

Difficulty:

Prerequisites: Good programming background in some language, and interest in Hanabi.

Jane's Projects

Art Gallery Problems. (Jane)

Description: Given a polygonal room, the art gallery problem asks how many guards one needs to place so that the whole room is guarded. That is, given that guards are stationary and can see 360 degrees in all directions, how many guards does one need to place so that they can see the whole room. Chvátal's art gallery theorem shows that an n-sided polygon can always be guarded with at most $\lfloor \frac{n}{3} \rfloor$ guards.

Since then, people have posed many variations of the art gallery problem, with some successfully solved and some still open. For example, we could ask if a similar theorem holds for the prison guard problem, where guards are placed outside the polygon and must see all of the walls. Or we could ask what happens in higher dimensions. Or we could ask for an algorithm to find the optimal placement of guards. Or we can have an art gallery where the walls are mirrors. The possibilities are endless!

In this project, we'll look into some of these variations (or come up with our own) and explore what is both known and unknown.

Structure:

Expected Input: A few hours a week, either in groups or individually

Expected Output: Understanding of what is currently known about some variation(s) of the art gallery problem, and an attempt at tackling the unknown

Difficulty:

Prerequisites: None.

Fractal Art. (Jane)

Description: Fractals are beautiful, but they can be hard to draw by hand, so let's draw them by computer! There are many different fractals and many ways to construct them, but some examples of fractals that we can explore are fractals created by recursion (e.g. Sierpinski's triangle, or the Koch curve), the Mandelbrot set, Julia sets, Apollonian circle packings, and tessellations of the hyperbolic plane.

This project is a chance for you to explore the math behind various fractals and to create some computer-generated fractal art.

Structure: Campers work individually or in groups to code fractal images.

Expected Input: Up to you

Expected Output: Pictures to show off at the project fair

Difficulty:

Prerequisites: A bit of coding experience would be helpful.

Jeff's Projects

Building a Radio. (Jeff)

Description: One of the oldest electric technologies is the radio. An AM (amplitude modulated) radio station transmits a function $f(t)$ on frequency ξ by sending the signal

$$\phi(t) - f(t)\sin(\xi \cdot t)$$

into the air. Someone who receives this signal then takes $\phi(t)$, and (because they're trying to listen to the ξ-frequency station) divides by $\sin(\xi t)$ to recover $\phi(t)$.
But ξ is not the only radio station that plays music. There is also station ζ, which transmit $\psi(t) = g(t)\sin(\zeta t)$. So, when someone listening to the radio wants to listen to station ξ, they actually receive

$$\phi(t) = \psi(t),$$

so how do they recover the original signal?
Once we develop a mathematical model for doing this, we are going to build an electronic circuit which actually implements this algorithm, and listen to the airwaves.
Structure:

Expected Input: Students will work with advisor to learn mathematics behind band-pass filters, then we will build a band pass filter and amplifying circuit.

Expected Output: A Radio!

Difficulty:

Prerequisites: You should know Ohm's Law, and Calculus.

Games with Graphs. (Jeff)

Description: The game of Brussel Sprouts is a game involving topological graphs. The playing field is a sheet of paper, with several x's drawn on it. On their turn, each player may take a point and connect it to another point by an edge that does not cross the other edges. They then add a little slash to the that edge, creating two new points. This process slowly builds up a planar graph, until one of the two players is unable to make a move and loses.
This game depends not only on the graph created, but on the topology of the drawing of this graph. Come and see me during the project fair if your interested in this game or other games with graphs!

Structure:

Expected Input: Students will work in a group mostly without supervision. 5–10 hours a week.

Expected Output: A complete solution to Brussel Sprouts, and a partial solution to Soy Sprouts.

Difficulty:

Prerequisites: Graph Theory

Lines and Knots. (Jeff)

Description: Ok, pretty obvious that every knot has a line that intersects it. In fact, it is also pretty clear that every knot has a line that intersects it at 2 points. Can you show that every knot (which is not the unknot) has a line that intersects it at 3 points? What about 4 points? What about 5 points?
Structure:

Expected Input: Working several hours a week towards finding a trisecant of quadresecants. If you get stuck, or need ideas, I'm happy to help out and provide readings throughout the project.

Expected Output: Show the existence of trisecants, and possibly higher secants.

Difficulty:

Prerequisites: Point set topology, group theory

Knots in the Campus. (Jeff + Chris)

Description: Jeff keeps a ball of yarn in his pocket as he walks around the campus. When he leaves his room in the morning, he ties one end of the yarn ball to the door of his room. He then lets the yarn unspool throughout the day, and whenever he crosses the yarn on the ground, he steps over it. What kind of knots can he make this way? Is it possible to make all possible knots this way?
Structure:

Expected Input: 5–10 hours a week, plus meeting with adviser once a week to check up on progress.

Expected Output: A poster at the project fair about the type of knots that can be made based on the structure of the campus.

Difficulty:

Prerequisites: None

Cities and Graphs. (Jeff + Pesto)

Description: Take your favorite city (like Waterville) and pick a few of your favorite destinations. The roads between the roads form a graph, whose edges you can weight by the travel time along the road.

When we draw a map of a city, we usually draw it so that the roads between those locations resemble the roads as drawn on the Earth. However, you could imagine drawing those locations so that all of the roads between them are straight lines whose lengths are the travel time between two points.

Is it always possible to do this? What types of conditions do we need to put on our graph to make this possible, and what kind of geometric properties do we learn about the graph from such a map?

Structure: Short weekly meetings.
Expected Input:

Expected Output:

Difficulty:

Prerequisites: None.

Mark's Projects

Knight's Tours. (Mark)

Description: Given a rectangular "chessboard" (of m by n squares), for what values of m and n can a knight traverse all the squares of the board (using legal moves) without ever visiting a square twice? (If you don't know how a knight moves in chess, ask.) For what values of m and n can this be done in such a way that the knight can leap back to the starting square from the ending square to create a "closed" knight's tour? In this project you'll be investigating these questions, and possibly (if you are dedicated and ingenious) answering them completely, or almost completely.

Structure: We'll meet to get you started, and then it's largely up to you.

Expected Input: At least several hours a week, but up to you. I can give hints and will be available during TAU, but I won't come after you if you don't persevere.

Expected Output: A poster for the project fair.

Difficulty:

Prerequisites: None, although you should be comfortable with mathematical induction, and you should not mind doing a lot of careful experimentation—if you dislike looking at many different cases, this is not the project for you.

Period of the Fibonacci sequence modulo n. (Mark)

Description: The Fibonacci sequence modulo 8 looks like

$$1, 1, 2, 3, 5, 0, 5, 5, 2, 7, 1, 0, 1, 1, \ldots ;$$

as you can see, it is periodic with period 12. In this project you'll investigate the period of the Fibonacci sequence modulo n as a function $f(n$; we just saw that $f(8) = 12$. This should lead to some ideas from number theory, as well as to an open problem (that is somewhat notorious and probably unreasonably difficult).

Structure: This would probably work best with 2 or 3 campers. Although I will be available for help during TAU and I'll be willing to give hints, I will not hunt you down if you don't keep working actively on the project, so you should only sign up for this one if you are highly motivated.

Expected Input: Up to you. In order to make serious progress, you should probably spend at least three or four hours per week, but it depends a bit on how quick you are.

Early on in the project there will be some computer/programmable calculator work so you get enough data to make conjectures. Once you have a table of enough values for the function, the rest of your work will be by hand.

Expected Output: A poster for the project fair.

Difficulty:

Prerequisites: A bit of number theory will be helpful, but as long as you know modular arithmetic you can work on the project and pick up the number theory as you go along. On the other hand, if you have had a serious college course on number theory (through the law of quadratic reciprocity) the project probably won't be as challenging as you might like. It would be good if at least one participant has a little programming experience - it doesn't really matter in what language or environment, as long as you know that you can get results on this campus.

Misha's Projects

How to Write a Cryptic Crossword. (Misha)

Description: Have you ever solved a cryptic crossword and thought, "I can do better than this"?

Do you look at the word "ADRIFT" and think it would make:

- a good charade clue?
- an anagram clue?
- an, I dunno, hidden word clue?

In this project, you will begin with an overview of different types of cryptic clues, and practice writing them for randomly chosen words. Then I'll introduce you to various tools that make writing a crossword easier, and you'll write a cryptic crossword of your own.

Structure: I encourage working in a team.

Expected Input: 3–4 hours a week of work.

Expected Output: A cryptic crossword at the project fair.

Difficulty:

Prerequisites: Familiarity with cryptic crossword clues. Attending one or more of Kevin and Sachi's cryptic crossword events may help with this.

How to Write an Olympiad Problem. (Misha)

Description: Writing problems for math competitions is hard, but watching people struggle with a problem you wrote is unexpectedly gratifying.

In this project, you will try to write your own math problem, which will appear on TPS in the last week of camp.

(For campers on a hardcore TPS team, I am willing to run a less rewarding version of this project: you will write a problem, but it will not appear on TPS.)

Structure: We'll adjust plans based on background, but I plan to give reading-and-thinking assignments to help you have ideas.

Expected Input: The amount of time necessary to write a good problem varies greatly. Expect to do a good bit of independent work.

Expected Output: A problem that will appear on TPS in week 5.

Difficulty: 🌶🌶🌶

Prerequisites: Familiarity with olympiad problem solving.

Willingness to take an oath of silence: you should not participate in week 5's Team Problem Solving, and you should not discuss the problem you write with other campers.

Nic's Projects

Algebraic Geometry Reading Group. (Nic)

Description: Form a small group of campers to read through Ravi Vakil's notes on algebraic geometry. What you focus on and how far you get is up to you!

Structure: After an organizational meeting with the group, you'll meet frequently to discuss the material and work on or present solutions to exercises. I'll be around for at least some of these meetings to help you out, but you'll be doing a lot of the work without me.

Expected Input: A lot of time spent reading, discussing the material with the other people in your group, and working on and discussing exercises.

Expected Output: A good understanding of whatever your group gets through before the end of camp!

Difficulty: 🌶🌶🌶🌶

Prerequisites: A thorough understanding of ring theory, probably beyond what was taught in Ari's class here, and a willingness to dig deep into some difficult material. Talk to me if you're unsure about whether you're ready.

Ray Tracing. (Nic)

Description: Ray tracing is a simple technique for producing beautiful 3D graphics. (Below this blurb you can see an example of the type of picture you can draw pretty easily with a ray tracer.) The basic idea is to figure out what color a pixel should be by following a ray of light coming into the camera through the scene to see what object it comes from.

If you decide to work on this project, you'll write a ray tracer of your own from scratch. A bare-bones ray tracer can be written without much code, but you can make it very complicated by adding features like shadows, focus, reflection, textures, and lots more.

Structure: You'll spend most of the time writing code, but I'll be around to answer questions or to talk about new features you want to add.

Expected Input: Plenty of quality time spent coding! I'll help you get started with producing graphics in your chosen language if you'd like.

Expected Output: A ray tracer that produces beautiful pictures to show off!

Difficulty: 🌶🌶

Prerequisites: Programming experience. Linear algebra might be helpful.

Non-Euclidean Video Games. (Nic, Chris)

Description: Have you ever wanted to play Tetris in a world where all parallel lines intersect? Play Asteroids in a world where the area of a circle grows exponentially with its radius? Write a video game that takes place in a space with a non-Euclidean metric!

Structure: We'll meet as a group to discuss game ideas and geometry, but after that you'll be expected to arrange meeting times on your own to get code written. We'll be available to consult with you about the programming and the geometry, and Nic can help you get started with a game engine in your favorite programming language.

Expected Input: Lots of quality time spent coding with your team.

Expected Output: A game!

Difficulty: 🌶🌶🌶

Prerequisites: Programming experience. Calculus might be helpful for a deeper understanding of the geometry, but it's not required.

Nic + Chris's Projects

Non-Euclidean Video Games. (Nic + Chris)

Description: Have you ever wanted to play Tetris in a world where all parallel lines intersect? Play Asteroids in a world where the area of a circle grows exponentially with its radius? Write a video game that takes place in a space with a non-Euclidean metric!

Structure: We'll meet as a group to discuss game ideas and geometry, but after that you'll be expected to arrange meeting times on your own to get code written. We'll be available to consult with you about the programming and the geometry, and Nic can help you get started with a game engine in your favorite programming language.

Expected Input: Lots of quality time spent coding with your team.

Expected Output: A game!

Difficulty: 🌶🌶🌶

Prerequisites: Programming experience. Calculus might be helpful for a deeper understanding of the geometry, but it's not required.

Pesto's Projects

Graph Minors Research. (Pesto)

Description: In Pesto's undergraduate thesis, he made one conjecture that he couldn't solve, but that he thinks might be approachable with no more than an hour's worth of graph theory background teaching. The statement:

For all positive integers k and all 2-connected simple graphs G with $|V(G)| \geq k+3$ and $\frac{|E(G)|-1}{|V(G)|-2} > 3 - \frac{1}{k}$, there exists a 2-connected simple graph H such that $G \geq H$, $|V(H)| \geq k+3$, and $|E(H)| \geq 3|V(H)| - 6$, where $\geq$ is minor containment.

Structure: Pesto'll teach the background at TAU one day (which people in the graph minors course can skip), and meet every other day or so to talk about any ideas you've had.

Expected Input: At least an hour to learn enough to understand the problems.

Expected Output: Minimum (1 hour): a bit of understanding of graph theory.

Medium (16 hours of thought, times or divided by 2): a solution to an easier version of the conjecture that Pesto solved.

Maximum (even if you put as much time as possible in and do well, you might not get anywhere): prove (or disprove) the conjecture.

Difficulty:

Prerequisites: Understand the terms used in the statement of the conjecture, except ”minor” and ”2-connected”.

Linguistics Problem Writing. (Pesto)

Description: You may have seen some linguistics logic puzzles, either on the relays or a contest like the North American Computational Linguistics Olympiad. For instance a small example: In Kurmanji Kurdish, the translations of {You see Bear, You see me, Bear runs, You run, I see Bear, I run} are, in some order, {Ez h'irç'e dibînim, Tu dir'evî, Tu min dibînî, H'irç' dir'eve, Ez dir'evim, Tu h'irç'e dibînî}. Which is which?

In this project, you'd learn some of the linguistics and math that go into making some such puzzles, then write such a puzzle.

Structure: Occasional meetings at TAU to talk about ideas for you to work on.

Expected Input: Time.

Expected Output: A linguistics logic puzzle, for relays or NACLO.

Difficulty:

Prerequisites: None.

Models of Computation Similar to Programming. (Pesto)

Description: A theoretical CS reading course, to answer questions like “why is finding a Hamiltonian cycle in a graph as hard as finding an elegant proof of any elegantly provable theorem?”

Structure: Reading course: you read, and occasionally we meet to discuss what youve read.

Expected Input: Reading time: the expected amount youll get out of it scales about linearly with the time you put in.

Expected Output: Some more complexity theory knowledge.

Difficulty:

Prerequisites: None.

Sachi's Projects

Arduino. (Sachi + Kevin)

Description: An arduino is a tiny computer that controls electronic circuits. We can connect it to LEDs, sensors, buzzers, motors, and more. We can build robots that draw, musical instruments triggered by light, and filter sound input to make you sound like your favorite mentor.

Structure: Students will work in groups of two to four. Projects may focus on any or all of programming, electronics, and design. All projects will involve elements of all three, but no previous experience in any of them is required.

Expected Input: Regular project meetings; more ambitious projects will require more time.

Expected Output: Something awesome powered by an arduino!

Difficulty: 🌶🌶

Prerequisites: Excitement for hands-on tinkering.

Sam's Projects

History of Math. (Sam)

Description: Interested in the History of Math? Taken one day of Moon's course and totally sold on the field? Then this is the project for you!

Logistics: we'll pick some part of the history of math that sounds interesting, and learn a bit about it. We can just read and chat about the stuff, or the project can end in some sort of short presentation in the academics lounge!

Possible directions: The St. Petersburg Paradox and the founding of probability, reading a paper by Alan Turing, reading something Cauchy wrote, etc.

Structure:

Expected Input:

Expected Output:

Difficulty: 🌶🌶

Prerequisites: None.

Modellin' Stuff (Statistically!) (Sam)

Description: Got a data set you'd like to attack? Have a question about camp you want to gather data to answer? In this project, you do just that!

Structure: Work on data analysis in R

Expected Input: 3–5 hours per week

Expected Output: Some knowledge about a data set!

Difficulty: 🌶🌶

Prerequisites: Statistical Modelling would be helpful

Reading Cauchy's Cours d'Analyse. (Sam)

Description: Cauchy effectively wrote the book on analysis; his Cours d'Analyse is a classic that's still relevant today. If this project were to run, a group of us would read through the first few chapters of his book and try to learn analysis according to Cauchy!

Structure: Depends on the number of campers interested. Will involve working through his book!
Expected Input: 3–4 hours/week
Expected Output:

Difficulty:
Prerequisites: Calculus would help

Zach's Projects

Build a Geometric Sculpture. (Zach)
Description: Design and build your own mathematical sculpture. The materials (cardboard? paper clips? chairs? yo-yos? Q-tips?) and underlying geometry (symmetric polyhedron? unstructured foam? Penrose tiling?) are unconstrained (within reason...) and will be decided together during the course of the project.

Structure: Regular meetings with Zach to discuss ideas, proposals, material purchases (if necessary), and progress.

Expected Input: Brainstorming and experimentation, leading to a final project design proposal

Expected Output: A finished, assembled sculpture

Difficulty:

Prerequisites: None.

Design an Origami Model. (Zach)
Description: Design an original origami model while learning some of the mathematical tricks of the trade from Robert Lang's "Origami Design Secrets". The complexity of the design is up to you, depending on your experience level and desired work amount.

Structure: Self-directed reading and experimentation; occasional meetings with Zach to review progress

Expected Input: Reading the text, experimenting to apply the techniques toward a desired model

Expected Output: A unique origami model of your own design

Difficulty:

Prerequisites: Some origami folding experience